

2006-01-2341

Environment-Aware Postural Control of Virtual Humans for Real-time Applications

Manuel Peinado, Daniel Meziat

Departamento de Automática. Universidad de Alcalá, Spain

Ronan Boulic, Daniel Raunhardt

VRLab. Ecole Polytechnique Fédérale de Lausanne, Switzerland

Copyright © 2006 SAE International

ABSTRACT

Interactive control of a virtual character through full body movement has a wide range of applications. However, there is a need for systems that accurately reproduce the motion of a performer while accounting for surrounding obstacles. We propose an approach based on an efficient Prioritized Inverse Kinematics constraint solver. The inputs to the system are the acquired positions of relevant body parts. These positions are tracked by the virtual character thanks to a set of kinematic constraints. At the same time a special constraint, which we call “observer”, is attached to body parts that must be checked for collisions. An observer acts by smoothly damping the motion of its controlled body part towards nearby obstacles, thus preventing future collisions from taking place.

INTRODUCTION

Interactive control of a virtual character through full body movement has applications in fields like virtual reality, workspace design or rapid prototyping of animations. Commercial systems are available that accurately capture human motion under the assumption that the virtual character is not surrounded by obstacles. However, if that assumption does not hold these systems would produce postures in which the character would penetrate its environment. Think, for instance, of a scenario in which the virtual character is standing in front of a wall. Any forward movement of the performer's arms should be corrected before being applied to the virtual character or else it would lead to a collision.

We propose a novel technique for real-time postural control of a virtual human in a cluttered environment. The inputs to the system are control signals corresponding to the position of selected body parts.

The motion of the performer is transferred in real-time thanks to a Prioritized Inverse Kinematics solver. Constraints are attached to the character to make it follow the position of the input signals coming from the sensors, while other constraints help reproduce the observed properties of the original motion, such as center of mass position or foot-to-ground relations.

In the key part of our system, a special type of constraint, which we call “observer”, is attached to each body part that must be checked for collisions. An observer acts by smoothly damping the motion of its controlled body part towards nearby obstacles, thus preventing future collisions from taking place. If this prevention mechanism fails, high priority constraints are created to solve the collision. The simultaneous satisfaction of these constraints and of those intended for matching the performer's posture make for a system in which postures are accurately transferred to the virtual character and corrected when necessary to avoid potential collisions.

RELATED WORK

Interactive control of a virtual control by means of direct transferral of motion is not new. In fact, its appeal to fields such as live TV production has turned it into a widely studied subject of research, usually referred to as *computer puppetry* [1,2]. Most systems require the use of extensive sensor or marker sets, which might render these systems too cumbersome or expensive for some applications. Less invasive solutions exist, often exploiting methods from the computer vision community, but they rarely enable control of the full body [3].

A common approach to the motion reconstruction problem is to limit the range of allowed actions to a predefined, application-specific, set. The problem then becomes one of finding which action in the database

best matches the current motion of the user. Although adaptable to a wide range of applications by adding new actions to the database, these techniques lack the flexibility required for a general-purpose motion reconstruction scenario [4,5].

Considerable effort has been devoted to the difficult problem of mapping motion from the performer to a target character of different size and proportion [2,6]. For simplicity, and throughout the rest of this paper, we will assume that in our system the dimensions of the target character match those of the performer (this can be enforced by modifying the skeleton of the target character during a calibration phase).

Of greater concern to the present work are methods that employ Inverse Kinematics to reconstruct the original motion of the performer. For performance reasons this methods usually divide the problem in different parts: first the body root is placed according to some optimization criteria, and then dedicated analytic IK solvers place the remaining body parts in order to solve the initial constraints [3,7,8]. Our method employs a different approach to Inverse Kinematics, since the whole body is taken into account for the satisfaction of each constraint [9].

The issue of real-time collision avoidance for high-dimensional articulated figures such as the human body has also received a good deal of attention from the Computer Graphics community. A recent trend advocates the use of commodity graphics hardware to accelerate collisions test. Redon et al., for instance, propose a very efficient method to detect collisions between a virtual character and its environment, but their response method is purely reactive [10]. Liu and Badler's effort in the same direction is able to plan collision-free reach motions in real time but allows only motion of the arm [11].

The idea of preventing collisions before they take place has also been explored to a good extent. Maciejewski and Klein were among the first to advocate this technique within a two priority layers architecture [12]. Another early work by Khatib achieves similar results by the use of artificial potential fields [13], whereas a more recent one by Garber and Lin integrates repulsive constraints whose strength is a function of the distance to the obstacle that originates them [14]. The problem with these approaches is that a body part is repulsed from an obstacle even if it is not moving in its direction. This often yields unnatural results when applied to the animation of humans.

The computational complexity of finding collision-free motions for virtual human has led many researchers to investigate the use of randomized methods. These methods search in configuration space for a path linking a start and a goal configuration. Kuffner et. al have applied them to a humanoid robot with 33 DOFs, but only for offline applications [15]. A similar approach by

Kallmann et al. is able to generate reach and grasp motions in real-time for a 22 DOF humanoid [16], but the range of available motions is restricted to those pre-computed in a roadmap [17]. In general, all path planning techniques have the weakness that the user is required to specify a final collision-free posture.

Another idea that has received some attention is the use of a global low-dimensional path to help avoid the well-known local minima problem that arises in local planning methods. Hybrid path planning techniques exist that use this idea to cope with the complexity of planning in high-dimensional configuration spaces, at near real-time rates [18,19]. Garber and Lin apply the same idea to animate a robot arm working in an industrial context [14].

Other approaches to collision-free animation for virtual humans include the work of Choi and Cremer, which pose the problem of obstacle awareness as the resolution of nonlinear inequality constraints [20], the work of Kuffner et al., which deals specifically with the problem of avoiding self-collisions [21], or the work of Pettre et al., which are able to plan complete locomotion sequences thanks to a randomized method complemented by a warping phase [22].

METHOD OVERVIEW

We present a method for the interactive control of an articulated figure in an environment populated with obstacles. The method is applicable to figures with an arbitrary number of degrees of freedom, although we mainly focus on the animation of humanoids.

The problem can be stated as follows: given an articulated figure, a description of its surrounding environment, and a set of input control signals corresponding to the position of relevant body parts, find in real-time a sequence of poses that follow the control signals while avoiding collisions with nearby obstacles.

The input control signals typically come from sensors or markers located on the body of a performer. Our system is independent of the specific methods by which these signals were acquired (typical methods include optical and magnetic motion capture). For simplicity, we will henceforth refer to these signals as *sensors*. One benefit of our method is that a very small number of sensors is needed. Although more sensors could be used for higher accuracy, we have found that good results can be achieved by using only five (at the wrists, ankles and head of the performer). Using a reduced sensor set has two main advantages:

- The performer is less encumbered and is thus able to move more freely.
- The overall cost of the system is lower.

For the purpose of this work we will take our control sensors from a well-known database of pre-recorded human motions [23].

Given this problem statement, the solution we propose is summarized by the block diagram of Figure 1. At the core lies a state-of-the-art Prioritized IK solver, which is able to satisfy a set of multiple Cartesian constraints according to a strict priority ordering [25].

The *motion reconstruction* module has the responsibility of tracking the position of the input sensors. Since these sensors change on each frame of the motion, the IK solver must be run on each frame to find an updated pose that matches the new position of the input sensors.

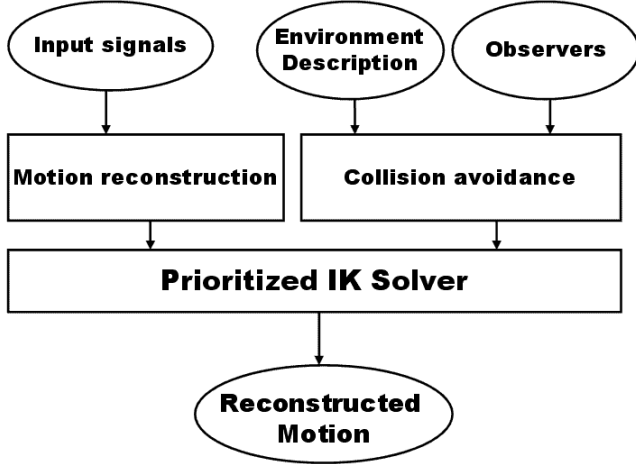


Figure 1: Architecture of the motion reconstruction system.

The motion reconstruction module is able to transfer the motion of the performer to the virtual character in real-time, but is unaware of its environment and may thus produce poses that penetrate surrounding obstacles. The *collision avoidance* module takes care of this by keeping a list of *observers*. An observer is a special type of entity that monitors the output of the IK solver on a frame by frame basis and activates a Cartesian inequality constraint when necessary. Each observer is attached to a part of the articulated figure, and has a shape that determines how it interacts with the environment.

An observer may influence the operation of the IK solver in two ways: *preventive* and *reactive*. The goal of the preventive mechanism is to anticipate immediate collisions and to prevent them from taking place. When an observer detects that its associate body part is moving towards a near obstacle, it activates a preventive inequality constraint aimed at damping the motion of the body part in the normal direction to the obstacle.

This prevention mechanism might fail, especially in presence of moving obstacles. When this happens an observer will detect that its body part is colliding with an obstacle and will enter reactive mode of operation. In this mode the observer activates a reactive constraint of high priority, whose goal is to solve the collision by pushing the body part away from the obstacle.

The IK solver, together with the observers that monitor its output constitute an efficient way to generate

collision-free sequences that satisfy the end-user constraints. In addition, the anticipation behaviour provided by the preventive inequality constraints give the method a sense of “intelligence” which allows it to avoid obstacles without waiting for a collision to take place.

MULTIPLE PRIORITY INVERSE KINEMATICS

The multiple Priority IK (also called Prioritized IK, or PIK) technique is used to reconstruct a believable posture of the user (i.e. its joint state, θ) given a task vector \mathbf{x} constructed from a set of user-specified Cartesian constraints (e.g., take joint to location or align joint with axis). In our motion reconstruction system, the task vector \mathbf{x} is created from:

- A set of predefined constraints to track the acquired position of the wrists, ankles and head.
- A set of dynamic constraints created by the collision avoidance module.

We give here a general overview of the PIK method, while the next section describes the specific set of constraints used in this paper, along with their relative priorities.

Our general architecture is based on the linearization of the set of equations expressing Cartesian constraints \mathbf{x} as functions of the joints’ degrees of freedom θ . By \mathbf{J} we denote the Jacobian matrix gathering the partial derivatives $\partial \mathbf{x} / \partial \theta$. We use its pseudo-inverse, denoted \mathbf{J}^+ , to build the projection operators $P_{N(J)}$ on the kernel of \mathbf{J} , noted $N(J)$. Our approach relies on an efficient computation of projection operators allowing to split the constraints set into multiple constraint subsets associated with an individual strict priority level [24].

The provided solution guarantees that a constraint associated with a high priority will be achieved as much as possible while a low priority constraint is optimized only on the reduced solution space that does not disturb all higher priority constraints. For example, such architecture is particularly suited for the off-line evaluation of reachable space by a virtual worker; in such a context the balance constraint is given the highest priority while gaze and reach constraints have lower priority levels [25].

Figure 2 provides an overview of our Prioritized Inverse Kinematics Architecture. The outer convergence loop is necessary as the linearization is valid only within the neighborhood of the current state; this requires to limit the norm of any desired constraint variation $\Delta \mathbf{x}$ toward their respective goal to a maximum value and to iterate the computation of the prioritized solution $\Delta \theta$ until the constraints are met or until the sum of the error reaches a constant value. This figure also highlights the clamping loop handling the inequality constraints associated to the mechanical joint limits. Basically we check whether the computed prioritized solution $\Delta \theta$ leads to the violation of

one or more joint limits. If it is the case, equality constraints are inserted to clamp the flagged joints on their limit and a new prioritized solution is searched in the reduced joint space.

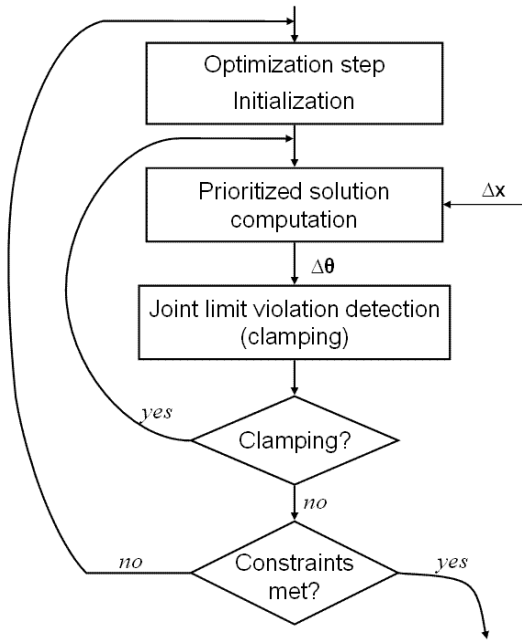


Figure 2: Outline of the constraint optimization loop integrating the Prioritized IK together with a clamping loop that enforces joint limits.

PRIORITIZED CONSTRAINTS FOR MOTION RECONSTRUCTION

As explained before, for each new motion frame, the control sensors provide us with information on the position of the performer’s wrists, ankles and head (although other sensor configurations are possible). It is necessary to use the Prioritized IK (PIK) solver to aid the motion capture process by consistently locating the joints that are not specified by the input sensors.

Associating a high priority with a constraint subset has the decisive consequence of guiding the convergence of the Prioritized IK through intermediate states that guarantee the enforcement of that specific constraint subset (as opposed to a weighted approach where intermediate states reflect a compromise solution between all constraints contributions). Such a channelling of the convergence through intermediate states satisfying at least the most important constraints is the first interesting property in a real-time context as we may not have the time budget for fully converging toward the maximum satisfaction of all the constraints.

Another interesting observation is the fact that the convergence through such type of constrained intermediate states also helps to avoid some local minima that are typical of weighted IK formulations [24]. It is hence very important to identify which constraint has the higher impact on the quality of the convergence and the visual appearance of the reconstructed posture. In our earlier experiments, limited to tracking the arms, the

highest priority was associated with the wrist location constraint while the shoulders and the elbow constraints were given respectively lower priority levels. We now describe the extended constraint set we consider in the present study.

As we address the posture identification of a standing person, the believability of the recovered posture is mostly governed by the correctness of its balance; the balance also has a great influence on the ability to reach a location in space [25]. For these reasons, we propose to exploit a skeleton able to model a simplified mass distribution of the whole body and to offer a control of the whole body center of mass. The prior observations on believability and reachability leads us to assign the highest priority to the centre of mass position constraint: this constraint ensures that the centre of mass projects over the root node to guarantee the balance.

Then the next most important constraint is the tracking of the ankle positions driven by the goal sensors. Attaining a good tracking of the ankle positions is important for the believability of reconstructed motions, especially during frames in which the character is standing (usually termed *ground* frames). In those frames, errors in the tracking of the ankles would lead to the character sliding on the floor. To avoid so, the ankle constraints are given the second highest priority after the center of mass. Immediately below in the priority ranking are the wrist-tracking and head-tracking constraints, which we deem less important because errors in their satisfaction lead to less noticeable artefacts.

Finally, we make use of a lowest-priority cost function optimization that attracts the current pose towards a predefined pose. We choose this predefined pose so that it has two important properties:

- The shoulder joints are in a configuration that helps keeping the arms away from the torso. This has the overall effect of reducing the likelihood of self-collisions involving the arms.
- The knees are bended. This helps the PIK solver generate, when necessary to satisfy other high-priority constraints, poses in which the character is crouched. We have found this to be of the utmost importance when reconstructing complex motions such as the crouch sequences shown in a subsequent section.

In addition to these two specific benefits, the optimization constraint adds to the quality of reconstructed motion by favoring configurations in which the spine and hip joints keep as close as possible to their initial rest values.

Table 1 summarizes this hierarchy of constraints, highlighting the different priority levels exploited in this study. Note that this table includes the collision-avoidance constraints that will be introduced in the next section.

In Figure 3 we demonstrate the effectiveness of this approach, by showing the result of reconstructing a jog motion and a vertical jump motion. The first one is shown accompanied by the original motion for the sake of comparison. In this experiment, as well as in those presented in later sections, we have used a skeleton composed of 18 joints (51 degrees of freedom).

Task	Priority
Center-of-mass control	1 (highest)
Track ankle sensors	2
Reactive collision-avoidance	3
Preventive collision-avoidance	4
Track wrists and head sensors	5
Cost function optimization	6 (lowest)

Table 1: Hierarchy of constraints and priority levels.

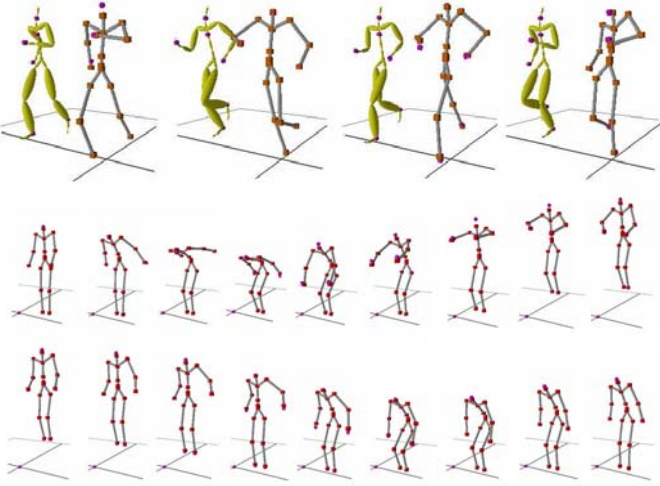


Figure 3: Reconstruction of two captured motions. Up: jogging (the left character shows the original motion.) Down: Jump vertically (for clarity, the original motion is not shown.)

COLLISION AVOIDANCE THROUGH THE OBSERVER PARADIGM

At the core level, motion in our system is generated by a Prioritized IK solver which is able to satisfy an arbitrary number of Cartesian constraints while respecting their relative priority order. Because this solver has no knowledge of the environment surrounding the virtual character, the configurations it generates will, in general, contain collisions. To address this issue we propose to add a collision-avoidance layer on top of the PIK solver. This layer is based on the key notion of *observer*. An observer is an entity that is attached to every body part that must be checked for collisions. The way an observer interacts with the environment depends on its shape. Currently this shape can be selected from a set of simple geometric primitives, such as line segment, sphere or cylinder.

On each iteration of the IK loop, observers check the position and velocity of their associate body part with

respect to surrounding obstacles. Depending on the result of this test, an observer may enter two different modes of operation: *preventive* and *reactive*. These are described in the following subsections.

PREVENTIVE OBSERVER ACTIVATION

The preventive mode of operation of an observer is triggered when its controlled body part:

- Is moving towards a near obstacle. Note that, in order to account for moving obstacles, the notion of *relative velocity* is used: the obstacle is assumed to be still (i.e., $\mathbf{v}_{\text{obst}}=0$) and the observer velocity is taken as $\mathbf{v}'_{\text{obs}} = \mathbf{v}_{\text{obs}} - \mathbf{v}_{\text{obst}}$ [26].
- Is inside the area of influence of the obstacle.

In this mode of operation, the observer activates a special damping constraint. As its name suggests, the role of this constraint is to damp the motion of the body part toward the obstacle (i.e., in the normal direction of the obstacle) while allowing full freedom in the tangent plane (see Figure 4). The damping factor is proportional to the distance to the obstacle, so that in the limit where the body part is touching the obstacle surface, no normal motion is allowed at all.

Figure 4 illustrates this concept on four points e_1 to e_4 moving with their respective position variations Δe_1 to Δe_4 . The displacement of e_1 is not altered because it always remains outside the area of influence. The displacement of e_4 is also not altered as it moves away from the obstacle. Only e_2 and e_3 see their displacement in the normal direction to the obstacle damped by a factor. This factor is bigger for e_3 because it is closer to the obstacle surface.

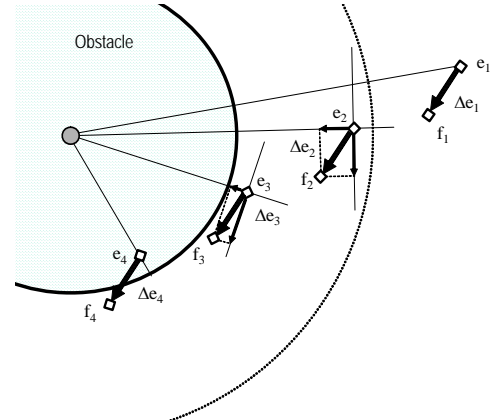


Figure 4: Damping in the influence area of an obstacle modeled as a spherical Cartesian inequality constraint. Only the displacements of e_2 and e_3 are altered.

The fact that we use the concept of relative velocity in this damping constraint has interesting implications. Think, for instance, of a scenario where an observer is not moving (thus $\mathbf{v}_{\text{obs}}=0$) but where an obstacle is moving towards it (with velocity \mathbf{v}_{obst}). One such scenario is shown in Figure 5. Because of our formulation in terms of the relative velocity, this will be interpreted just as if the obstacle was still and the observer was moving

in its direction. Thus, a preventive constraint will be activated by the observer. Here the result will not be a *damping* of the observer's motion as in Figure 4, (after all, the observer is not moving), but rather that the observer will be pushed away from the obstacle before an actual collision gets to take place. This can be appreciated in Figure 5, where a very simple 4 DOF chain changes its configuration due to the preventive constraints induced by an approaching obstacle.

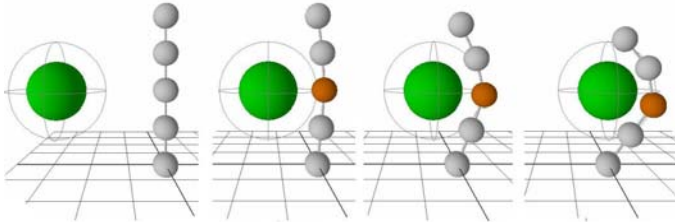


Figure 5: The moving obstacle (big sphere with surrounding influence area) induces the deformation of the articulated chain.

A big obstacle may trigger the activation of several preventive constraints at the same time. If all of them are given the same priority, the IK solve will strive to solve them simultaneously. Instead, we have found that better results are obtained by applying an *observer prioritization* policy: whenever several observers are influenced by an obstacle, only the closest one will activate a preventive constraint. The rationale behind this policy is that the closest body part will generally be more likely to lead to a future collision, and thus should be treated preferentially. Similar results are achieved by assigning higher priorities to closer observers, but activating only one constraint is preferable because a lower computation burden is imposed upon the IK solver. Note that this discussion applies only to observers activated by the same obstacle; two observers which have been activated by different obstacles will each activate its own damping constraints (see Figure 6).

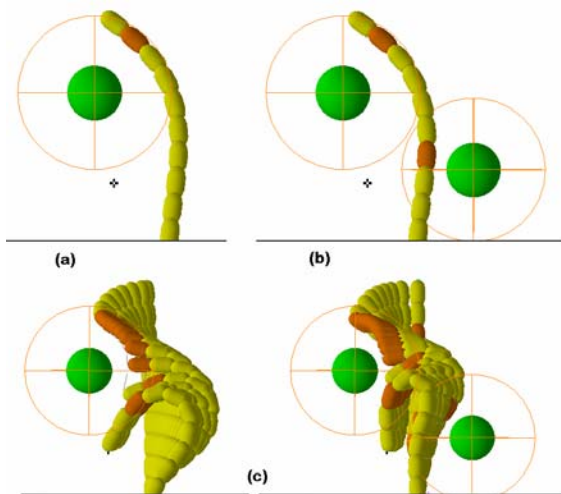


Figure 6: A 10 DOF kinematic chain reaching for a low goal. (a) Even though several observers are moving towards the obstacle, only the closest activates a preventive constraint. (b) Several observers can be activated by different obstacles. (c) Sequences generated for both examples.

REACTIVE OBSERVER ACTIVATION

The preventive mechanism might fail, generally owing to a moving obstacle that cannot be avoided by damping constraints. In that event, one or several body parts will be colliding with an obstacle. This is checked for on every IK iteration by the collision avoidance module. When the collision is detected, the observer enters its *reactive* mode of operation, activating a high-priority constraint whose goal is to push the body part away from the obstacle. Reactive constraints must be assigned a higher priority than preventive constraints because their mission is more demanding. That aside, we assign the same priority to all reactive constraints, so that if more than one is active at a given frame, it is up to the IK solver to satisfy both of them as much as it can.

HANDLING SELF COLLISIONS

Since each observer is assigned a shape so that it can be tested for collisions with nearby obstacles, observers themselves can be regarded as obstacles. This leads to a straightforward extension of our system to deal with self-collisions. Each time an observer tests for collisions with the environment, it treats the other observed body parts as obstacles and applies the same prevention and reaction mechanisms described above. Only two additional considerations are applied:

- Depending on the way we model the geometry of the virtual character, some observers may be in constant, unavoidable collision. Think, for example, of collisions between the upper and lower arm at the region near the elbow. In such cases, and adapting an idea introduced in [21], before the motion reconstruction begins we run an initialization procedure in which those observers which already collide on the rest pose are discarded from further collision tests.
- Checking for self-collisions introduces a significant computation burden due to the large amount of collision tests between observers that have to be performed. To alleviate this burden, our system extends the description of a virtual character with information on which body parts should not be checked against each other. This prevents a lot of unnecessary tests between body parts that are quite unlikely to collide, like the torso and the legs [21].

EXPERIMENTAL RESULTS

To isolate this study from the intricacies of a specific motion acquisition technique, we have tested the effectiveness of our system using pre-recorded motions from a database which is publicly available on the Web [23]. This database consists of several dozens of different motions recorded on different performers. We will concentrate on one particular motion, made of over 2200 frames, which contains several cycles of reach/crouch actions (see Figure 7). We chose this motion to test our system because it is challenging in that it involves the whole body.

Let us consider first in our first experiment an obstacle-free environment. This enables us to test how good the system performs when only self-collisions are taken into account. The result is shown in Figure 7. Eight line-shaped observers have been attached to the virtual character in order to generate this demo: 4 for the upper limbs and 4 for the lower limbs. The same observer settings will be used in the demos presented next, except if otherwise stated.

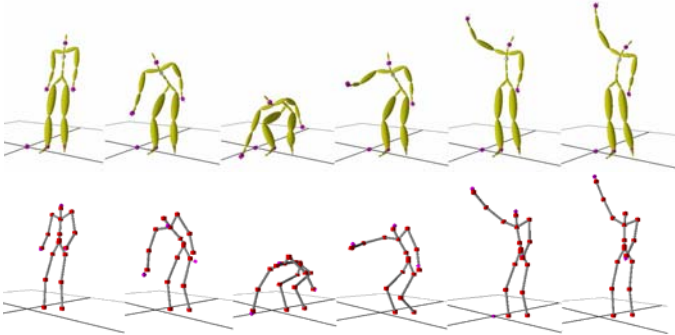


Figure 7: Part of the original crouch/stand-up sequence (top row) and motion reconstructed in a obstacle-free environment (bottom row)

The little boxes in the bottom row of Figure 7 represent the position of the input sensors, and their proximity to the associate body part gives a hint of the quality and responsiveness of the motion reconstruction. It can be noted how, as a result of the priority hierarchy described on a previous section, the ankles are perfectly tracked, whereas the wrists and head, constrained with a lower priority, experience small tracking errors. These tracking errors can be reduced at will by increasing N_{IK} , the number of IK iterations per motion frame. Of course, a compromise value of N_{IK} must be adopted because it has a direct impact on the system performance. We've used $N_{IK}=5$ for all the examples in this paper.

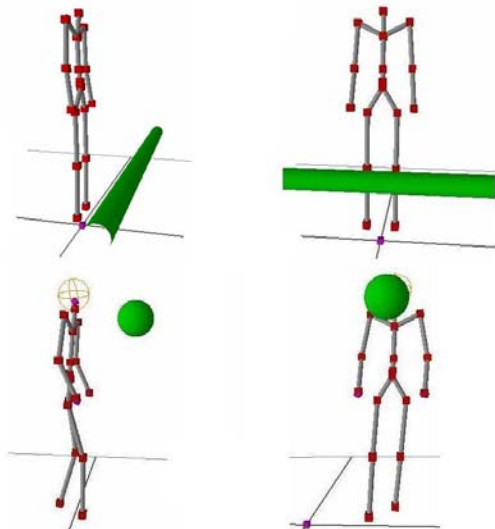


Figure 8: Obstacle setup for experiments 1 (top row) and 2.

In the next test scenario (top of Figure 8) we add a cylindrical obstacle which stands in the way of the right hand during some of the crouch cycles. The obstacle is assigned a large area of influence to obtain a good sense of anticipation to the reconstructed motion. Figure 9 shows some frames from the resulting animation. It

can be seen how, as it moves its arm up and down, the virtual character carefully avoids colliding with the cylinder.

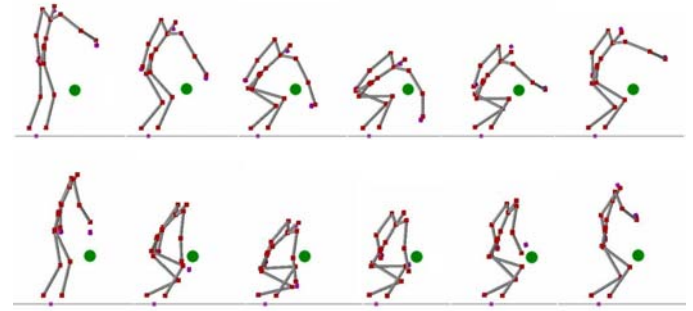


Figure 9: Two sample sequences from different parts of the reconstructed motion of experiment 2.

In our final experimental setting (Figure 8, bottom) a spherical obstacle is added at a position such that it hinders the motion of the virtual character's head as it stands up back from a crouch action. This is challenging because the motion of the whole body has to be rectified in order to avoid the obstacle, not just the motion of the arms as in the former scenario. The head itself is modelled as a spherical observer which is attached to the neck joint. As shown in Figure 10, the virtual character stands up as desired while avoiding the spherical obstacle thanks to a coordinated motion of the whole body.

Table 2 summarizes the frame rates measured for our three experimental scenarios. They are all averaged over all frames in the sequence (2200). We concentrate on the frames-per-second (fps) parameter because it is a direct indication on the applicability of the system in real-time. A fps of 20-25 would be enough to track most input motions without losing any important features. Table 2 shows that our system fulfills this requirement by a good margin. Experiments 2 and 3 are quite similar in their performance, the later being slightly slower because of the extra spherical observer needed to represent the head.

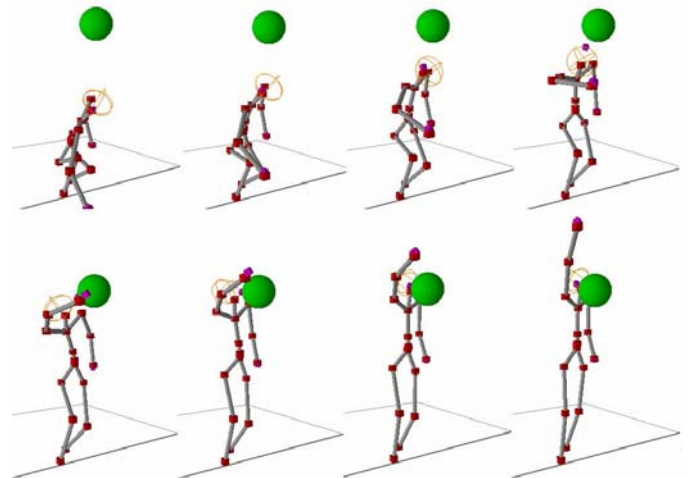


Figure 10: A sample sequence from a stand-up cycle of experiment 3.

CONCLUSION

Prioritized Inverse Kinematics is a powerful technique that eases the task of reconstructing the motion of complex articulated figures, such as virtual humans, from a reduced set of control parameters. We have presented an effective way that extends this technique to environments populated with obstacles. Two modules work together to provide environment-aware motion reconstruction capabilities. In the first module a set of Cartesian constraints, sorted according to a strict priority ranking, are enforced in order to track the known features of the original motion (wrist, ankle and head positions).

Experiment	Fps
1 (no obstacles)	52.6
2 (cylindrical obstacle)	42.5
3 (spherical obstacle)	40.6

Table 2: System performance in our three experiments. Timings were measured on a 1.8 GHz AMD Sempron machine with 1 GB RAM. Rendering time is not included.

In the second module, several entities called observers are attached to relevant body parts and monitor their motion, activating preventive and reactive constraints that anticipate future collisions or solve collisions that have already taken place.

Experimental evaluations of the different modules have been presented for a human crouch/stand example motion. The motion reconstruction capabilities of the system have been demonstrated, as well as its ability to anticipate future collisions thanks to the prevention mechanism. Experimental results have been provided that show that the system is suitable for real-time and interactive applications.

FUTURE WORK

In our current approach the constraints used to track the sensors have fixed priorities. In particular, as explained in a previous section the ankle constraints have a high priority in order to avoid the foot-slide problem. This makes sense only during ground frames (i.e., frames in which the character stands with the feet on the ground). For flight frames, or frames of partial support, a different assignment of priorities might lead to better results or to faster convergence. We plan to investigate this in order to obtain a dynamic, action specific, priority assignment mechanism.

We also plan to integrate our past work on motion strategies [24] into our motion reconstruction system. This will enable us to exploit the characterization of the virtual character's reachable space to select different action strategies at runtime.

ACKNOWLEDGEMENTS

This research has been partially supported by the the Swiss National Science Foundation under grant 200020-109989 and by the University of Alcalá under grant PI2005/083.

REFERENCES

1. Sturman, D.J.: Computer Puppetry, *IEEE Comput. Graph. Appl.* 18(1), 38–45, 1998.
2. Shin, H.J., Lee, J., Shin, S.Y., Gleicher, M.: Computer puppetry: An importance-based approach. In *ACM Trans. Graphics.* 20(2), 67–94, 2001.
3. Badler, N.I., Hollick, M.J., Granieri, J.P.: Real-Time Control of a Virtual Human Using Minimal Sensors, In *Presence* 2, 82–86, 1993.
4. Lee, J., Chai, J., Reitsma, P.S.A., Hodgins, J.K., Pollard, N.S.: Interactive control of avatars animated with human motion data. In *Proc. Of SIGGRAPH '02*, pp. 491–500, 2002.
5. Sidenbladh, H., Black, M.J., Sigal, L.: Implicit Probabilistic Models of Human Motion for Synthesis and Tracking. In *European Conference on Computer Vision (ECCV)*, pp. 784-800, 2002.
6. Gleicher, M.: Retargetting motion to new characters. In *SIGGRAPH '98*, ACM Press, pp. 33–42, 1998.
7. Boulic, R., Varona, J., Unzueta, L., Peinado, M., Suescun, A., Perales, F.: Real-Time IK Body Movement Recovery from Partial Vision Input. In *Proc. of the 2nd International Enactive Workshop*, Genoa, November 2005.
8. Semwal, S.K., Ron, R.H., Stansfield, S.: Mapping algorithms for real-time control of an avatar using eight sensors. In *Presence*, 7(1), 1–21, 1998.
9. Peinado, M., Herbelin, B., Wanderley, M., Le Callennec, B., Boulic, B., Thalmann, Méziat, D.: Towards Configurable Motion Capture with Prioritized Inverse Kinematics. In *Proc. of the third International Workshop on Virtual Rehabilitation (IVWR'04)*. Lausanne, Sept. 2004.
10. Redon S., Kim Y. J., Lin M. C., Manocha D., 2004, Fast continuous collision detection for articulated models. In *Proceedings of ACM Symposium on Solid Modeling and Applications*, 2004.
11. Liu, Y., Badler, N.I.: Real-time Reach Planning for Animated Characters Using Hardware Acceleration. In *Proc. of Computer Animation and Social Agents (CASA'03)*, pp. 86-93, 2003.
12. Maciejewski A.A., Klein, C.A.: Obstacle Avoidance for Kinetically Redundant Manipulators in Dynamically Varying Environments. In *Int'l J. Robotics Research*, 4(3), pp. 109-117. 1985.
13. Khatib, O.: Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. In *IEEE International Conference on Robotics and Automation*, St. Louis, Missouri, pp. 500-505. 1985.

14. Garber, M., Lin, M.C.: Constraint-based Motion Planning Using Voronoi Diagrams. In *Proc. of Workshop on Algorithmic Foundations of Robotics*, 2002.
15. Kuffner, J.J., Nishiwaki, K., Kagami, S., Inaba, M., Inoue, H.: Motion planning for humanoid robots. In *Proc. 20th Int'l Symp. Robotics Research (ISRR'03)*, Italy, October 2003.
16. Kallmann, M., Aubel, A., Abaci, T., Thalmann, D.: Planning Collision-Free Reaching Motions for Interactive Object Manipulation and Grasping, *Computer Graphics Forum* 22(3) pp. 313-322, 2003.
17. Kuffner, J.J., LaValle, S.M.: RRT-Connect: An Efficient Approach to Single-query Path Planning. In *Proc. of Int'l Conf. on Robotics and Automation (ICRA'00)*, 2000.
18. Brock, O., Khatib., O.: Elastic Strips: A Framework for Motion Generation in Human Environments. In *Int'l J. Robotics Research*, 21(12), pp. 1031-1052, 2002.
19. Brock, O., Kavraki, L.: 2001. Decomposition-based Motion Planning: A Framework for Real-time Motion Planning in High-dimensional Configuration Spaces. In *IEEE Int'l Conf. on Robotics and Automation*, 2001.
20. Choi, M.H., Cremer, J.F.: Geometric Awareness for Interactive Object Manipulation. In *Graphics Interface*, pp. 9-17, 1999.
21. Kuffner, J., Nishiwaki, K., Kagami, S., Kuniyoshi, Y., Inaba, M., Inoue, H.: Self-collision Detection and Prevention for Humanoid Robots. In *Proc. of Int'l Conf. on Robotics and Automation (ICRA'02)*, pp. 2265-2270, 2002.
22. Pettre, J., Laumond J.-P., Simeon T.: A 2-Stages Locomotion Planner for Digital Actors. In *Proc. of Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, pp 258-264, 2003.
23. The CMU Graphics Lab Motion-Capture Database. <http://mocap.cs.cmu.edu/>
24. Baerlocher, P., Boulic, R.: An Inverse Kinematic Architecture Enforcing an Arbitrary Number of Strict Priority Levels. In *The Visual Computer*, Springer Verlag, 20(6), pp 402-417, 2004.
25. R. Boulic, R., Baerlocher, P., Rodríguez, R., Peinado, M., Meziat, D.: Virtual Worker Reachable Space Evaluation with Prioritized Inverse Kinematics. In *35th International Symposium on Robotics*, 2004.
26. Fiorini, P., Shiller, Z.: Motion Planning in Dynamic Environments Using the Relative Velocity Paradigm. In *Proc. of the 1993 IEEE Int. Conf. on Robotics and Automation*. Vol. 1, pp. 560-565, 1993.